IN THE CLAIMS

1.   (Currently Amended) A method for speculatively reusing regions of code, the method comprising:

identifying a reuse region and ~~a data~~ input data [[to]] of the reuse region;

searching ~~a buffer storing~~ a plurality of instances of the reuse region for a valid matching instance that has [[a]] matching input data ~~input~~ and cannot be potentially invalidated; and

if the valid matching instance is not found, predicting, for the reuse region, a current set of live-out ~~registers and an output value for each live-out register in the set~~ values using reuse region instance information ~~in the buffer~~ pertaining to the plurality of instances of the reuse region.

2.   (Currently Amended) The method of claim 1 wherein searching the ~~buffer~~ the plurality of instances comprises:

determining whether the ~~data~~ input data [[to]] of the reuse region matches any input information within the reuse region instance information; and

if the ~~data~~ input data matches input information within the reuse region instance information, determining whether the reuse region is identified by a normal reuse instruction.

3.   (Original) The method of claim 1 wherein the reuse region instance information includes input information and output information for each instance of the reuse region.

4.   (Currently Amended) The method of claim 3 wherein the reuse region instance information further includes a plurality of confidence counters for each live-out register of the reuse region, each of the plurality of confidence counters being associated with a ~~certain~~ prediction technique.

5.   Cancelled.

6.   (Currently Amended) The method of claim 4 wherein predicting <u>the current set of live-out values</u> ~~an output value for each live-out register~~ comprises:

comparing a plurality of confidence ~~counters~~ <u>counts</u> associated with ~~said~~ each live-out ~~register~~ <u>value in the set</u>;

selecting an optimal prediction technique for said each live-out ~~register~~ <u>value in the set</u> based on comparison; and

selecting [[the]] <u>an</u> output value for said each live-out ~~register~~ <u>value in the set</u> using the optimal prediction technique.

7.   (Previously Presented) The method of claim 6 wherein the optimal prediction technique is any one of a context-based prediction technique, a stride prediction technique, and a last value prediction technique.

8.   (Currently Amended) An apparatus comprising:

a buffer to hold reuse region instance information pertaining to a plurality of instances of a reuse region; and

a processing core to search the buffer for a valid matching instance that has ~~a data~~ input data matching [[a]] current ~~data~~ input data of the reuse region and cannot be potentially invalidated, to predict for the reuse region a current set of live-out registers and an output value for each live-out register in the set based on the reuse region instance information if the valid matching instance is not found in the buffer, and to speculatively execute instructions using predicted output values of the reuse region.

9.   Cancelled.

10.   (Currently Amended) The apparatus of claim 9 wherein the processing core is to search the buffer for a valid matching instance by finding an instance with [[data]] input data matching the current [[data]] input data of the reuse region and determining whether the reuse region is identified by a normal reuse instruction.

11.   (Original) The apparatus of claim 8 wherein the reuse region instance information includes input information and output information for each instance of the reuse region.

12.   (Currently Amended) The apparatus of claim 11 wherein the reuse region instance information further includes a plurality of confidence counters for each live-out register of the reuse region, each of the plurality of confidence counters being associated with a ~~certain~~ prediction technique.

13.   (Currently Amended) The apparatus of claim 8 wherein the buffer includes a prediction list having a plurality of pointers to reuse region instances held in the buffer, a

pointer to the most currently used instance being located [[on]] at the top of the prediction list and a pointer to the least currently used instance being located at the bottom of the prediction list.

14.   (Original) The apparatus of claim 8 wherein the buffer includes a value prediction table having an entry that includes a predicted output value, the predicted output value being located using an index.

15.   (Previously Presented) The apparatus of claim 13 wherein the processing core is to predict an output value for each live-out register by comparing a plurality of confidence counters associated with said each live-out register, selecting an optimal prediction technique for said each live-out register based on comparison, and selecting the output value for said each live-out register using the optimal prediction technique.

16.   Cancelled.

17.   (Previously Presented) The apparatus of claim 13 wherein:

the prediction list points to a most recently used instance if the optimal prediction technique is a last value prediction technique;

the prediction list points to two most recently used instances if the optimal prediction technique is a stride prediction technique; and

the prediction list points to instances associated with a corresponding live-out register if the optimal prediction technique is a context-based prediction technique, the

associated instances being used to calculate an index pointing to a predicted output value in a value prediction table maintained in the buffer.

18.   (Currently Amended) A system comprising:

a memory to store regions of code; and

a processor, coupled to the memory, to identify a reuse region in the regions of code, to search ~~a buffer storing~~ a plurality of instances of the reuse region for a valid matching instance that has [[a]] matching input data ~~input~~ and cannot be potentially invalidated, and to predict for the reuse region a current set of live-out ~~registers and an output value for each live-out register in the set~~ values using reuse region instance information ~~in the buffer~~ pertaining to the plurality of instances of the reuse region if the valid matching instance is not found.

19.   Cancelled.

20.   (Previously Presented) The system of claim 18 wherein the reuse region instance information includes input information and output information for each instance of the reuse region.

21.   (Currently Amended) The system of claim 18 wherein the reuse region instance information includes a plurality of confidence counters [[for]] associated with each live-out value in the set ~~register of the reuse region~~, each of the plurality of confidence counters being associated with a ~~certain~~ prediction technique.

22.   Cancelled.

23.   Cancelled.

24.   (Currently Amended) A ~~computer readable~~ machine accessible medium comprising instructions, which when executed on a ~~processor~~ processing system, perform a method for speculatively reusing regions of code, the method comprising:

identifying a reuse region and ~~a data~~ input data [[to]] of the reuse region;

searching ~~a buffer storing~~ a plurality of instances of the reuse region for a valid matching instance that has [[a]] matching input data ~~input~~ and cannot be potentially invalidated; and

if the valid matching instance is not found, predicting, for the reuse region, a current set of live-out ~~registers and an output value for each live-out register in the set~~ values using reuse region instance information ~~in the buffer~~ pertaining to the plurality of instances of the reuse region.

25.   (Currently Amended) The ~~computer readable~~ machine accessible medium of claim 24 wherein searching the ~~buffer~~ plurality of instances of the reuse region comprises:

determining whether the input data ~~input to~~ of the reuse region matches any input information within the reuse region instance information; and

if the ~~data~~ input data matches input information within the reuse region instance information, determining whether the reuse region is identified by a normal reuse instruction.

26.   (Currently Amended) The ~~computer readable~~ <u>machine accessible</u> medium of

claim 24 wherein the reuse region instance information includes input information and

output information for each instance of the reuse region.

27.   (Currently Amended) The ~~computer readable~~ <u>machine accessible</u> medium of

claim 26 wherein the reuse region instance information further includes a plurality of

confidence counters [[for]] <u>associated with</u> each live-out ~~register of the reuse region~~ <u>value</u>

<u>in the set</u>, each of the plurality of confidence counters being associated with a ~~certain~~

prediction technique.

28.   Cancelled.

29.   (Currently Amended) The ~~computer readable~~ <u>machine accessible</u> medium of

claim 27 wherein predicting [[an]] <u>the current set of</u> output ~~value~~ <u>values</u> ~~for each live-out~~

~~register~~ comprises :

        comparing [[a]] <u>the</u> plurality of confidence counters ~~associated with said each~~

~~live-out register~~;

         selecting an optimal prediction technique for said each live-out ~~register~~ <u>value in</u>

<u>the set</u> based on comparison; and

        selecting [[the]] <u>an</u> output value for said each live-out ~~register~~ <u>value in the set</u>

using the optimal prediction technique.

30.   (Currently Amended) The ~~computer readable~~ <u>machine accessible</u> medium of claim 29

wherein the optimal prediction technique is any one of a context-based prediction technique,

a stride prediction technique, and a last value prediction technique.